**OPTIMAL BINARY SEARCH TREE**

#include<stdio.h>

#include <limits.h>

int n;

int Find(int c[n+1][n+1],int r[n+1][n+1],int i,int j) {

int min = INT\_MAX, l, m;

for ( m = r[i][j-1] ;m<=r[i+1][j]; m++ )

if (c[i][m-1] +c[m][j] < min ) {

min=c[i][ m-1] +c[m][j];

l=m;

}

return l;

}

void OBST(int p[],int q[],int n) {

int i,m,k,j,w[n+1][n+1],c[n+1][n+1],r[n+1][n+1];

for (i=0 ; i<=n-1;i++) {

w[i][i]=q[i];

r[i][i]=0;

c[i][i]=0;

w[i][i+1]=q[i]+q[i+1]+p[i+1];

r[i][i+1]=i+1;

c[i][i+1]= q[i]+q[i+1]+p[i+1];

}

w[n][n]=q[n];

r[n][n]=0;

c[n][n]=0;

for(m=2;m<=n;m++) {

for( i=0;i<=n-m;i++) {

j = i + m;

w[i][j]=w[i][j-1]+p[j]+q[j];

k = Find(c,r,i,j);

c[i][j]=w[i][j]+c[i][k-1]+c[k][j];

r[i][j]= k;

}

}

printf("\n\nWeight matrix : \n");

for(i=0;i<=n;i++){

for(j=0;j<=n;j++) {

if( i<=j){

printf("%2d ",w[i][j]);

}

else{

printf(" ");

}

}

printf("\n");

}

printf("\n\nCost matrix : \n");

for(i=0;i<=n;i++) {

for(j=0;j<=n;j++) {

if( i<=j){

printf("%2d ",c[i][j]);

}

else {

printf(" ");

}

}

printf("\n");

}

printf("\nRoot matrix : \n");

for(i=0;i<=n;i++) {

for(j=0;j<=n;j++) {

if( i<=j){

printf("%2d ",r[i][j]);

}

else {

printf(" ");

}

}

printf("\n");

}

}

int main() {

int i;

printf("Enter value of n : ");

scanf("%d",&n);

int p[n+1],q[n+1];

printf("Enter value of Array P : ");

for(i=1 ;i<=n;i++)

scanf("%d",&p[i]);

printf("Enter value of Array Q : ");

for(i=0 ;i<=n;i++)

scanf("%d",&q[i]);

OBST(p,q,n);

return 0;

}

